

N72-12117  
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

*Technical Memorandum 33-516*

*DALG - A Program for Test Pattern  
Generation in Combinational  
Logical Circuits*

CASE FILE  
COPY

JET PROPULSION LABORATORY  
CALIFORNIA INSTITUTE OF TECHNOLOGY  
PASADENA, CALIFORNIA

November 15, 1971

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

*Technical Memorandum 33-516*

*DALG – A Program for Test Pattern  
Generation in Combinational  
Logical Circuits*

JET PROPULSION LABORATORY  
CALIFORNIA INSTITUTE OF TECHNOLOGY  
PASADENA, CALIFORNIA

November 15, 1971

## PREFACE

The work described in this report was performed by the Quality Assurance and Reliability Division of the Jet Propulsion Laboratory.

1973-1974

## CONTENTS

Introduction . . . . .	1
A. Test Pattern Generation . . . . .	1
B. Input . . . . .	2
C. Verification of Test Patterns . . . . .	2
D. Input Error Detection . . . . .	3
E. Limitations . . . . .	3
DALG — A Program for Test Pattern Generation in Combinational Logical Circuits . . . . .	4
Input . . . . .	6
Output . . . . .	10
Description . . . . .	14
References . . . . .	15
Sample Problem, Circuit No. 1 . . . . .	16
Sample Problem, Circuit No. 2 . . . . .	17
Appendix . . . . .	37

## ABSTRACT

The document is primarily a user's manual for a computer program DALG which generates test patterns for detecting faults in combinational logic circuits containing up to 200 logical gates. The gates may be of logical types AND, OR, NAND, NOR, NOT, or Exclusive OR. The faults may be any one gate or input stuck at a fixed value (0 or 1).

In addition to test pattern generation DALG will also determine whether or not the given test pattern will detect given faults in a circuit. Sample problems are given along with input data sheets and printed output to illustrate the capabilities of the program.

## I N T R O D U C T I O N

The material contained in this section of this document is designed to describe briefly, in broad terms, the uses of the computer program DALG. The remainder of the document is designed to guide a user of the DALG program.

The program DALG will generate test patterns to detect faults in combinational logical circuits containing up to 200 logical gates. The gates may be of logical types AND, OR, NAND, NOR, NOT, or EXCLUSIVE OR. The faults may be any one gate or input stuck at a fixed value (0 or 1).

In addition to the sample problems used in the document, an application to a large scale integrated circuit is shown in an appendix.

The program has the following features:

### A. Test Pattern Generation

1. A request to generate a test pattern for the detection of a specific fault in a circuit may be made.
2. A request for a sequence of test patterns to detect all possible simple faults in a circuit requires only one card (simple fault - a single gate or input stuck at 0 or stuck at 1).
3. A test pattern generated to detect a particular fault is checked by simulation against a list of faults to be detected and, if the test pattern is capable of detecting additional faults, this fact is noted.

4. As test patterns for detection of faults are generated, these faults are automatically deleted from the list of faults to be detected. This minimizes to some extent the number of test patterns required to screen for all possible simple faults.
5. For a given test pattern, the status of each logical gate and input in the failure-free circuit is printed out. The same information is also printed out for the failing circuit in which the given test pattern will detect the failure.

B. Input

1. A circuit to be analyzed by DALG is described in terms of inputs and logical gates and:
  - a. the gate or input is denoted by a pair of Hollerith symbols;
  - b. each gate is described on a single card as to its logical function and its inputs.

These properties allow easy modification of an input deck when a circuit is changed.

2. Simple control cards allow the user to easily control the order of analysis on circuits.

C. Verification of Test Patterns

Arbitrary test patterns may be checked to determine whether or not they will detect a given fault in a circuit.

1. The test patterns may originate as input or from the test pattern generation portion of DALG.

2. The faults simulated may be multiple faults and may be specified as input or be pseudo randomly generated faults.

D. Input Error Detection

The program performs extensive error checking on the input data.

E. Limitations

The program at present is limited to circuits with at most 40 inputs and 200 logical gates, but these limits can be increased with only minor changes in the program.

DALG  
A PROGRAM FOR TEST PATTERN GENERATION  
IN COMBINATIONAL LOGICAL CIRCUITS

---

The program DALG will generate test patterns to detect faults in combinational logical circuits. The algorithm considers only logical faults of gates or inputs and only those faults stuck at 0 or stuck at 1. A test pattern which will detect a specific fault is a set of input values such that the value of at least one output for the fault free circuit is different than that of the circuit with the specific fault.

The program is based upon an algorithm developed by J. P. Roth of IBM (Ref. 1). This algorithm functions by tracing out the effects of a logical failure through only those paths which are "sensitive" to the failure. If a so-called "sensitized path" contains an output, then the remainder of the circuit is checked to see if there is a set of values for the (internal) outputs of each gate and a set of inputs which are consistent with logical values at every point along the sensitized path. If this is the case, then a test for a fault is found. If, on the other hand, inconsistencies are found for all possible values which can be varied, then another sensitized path is chosen and the procedure repeated. Either a test pattern will be found or all paths in the circuit sensitive to the specific fault will be exhausted. In this latter case, there is no test for the fault and, in fact, there is redundancy in the circuit; for greater detail on the algorithm, see the monograph by Chang, Manning and Metze (2) and (1) by Roth.

In order to facilitate use of the program, a brief of its major parts is in order. They are 1) DALG, 2) CREAD, 3) FTEST, 4) OPUT, and 5) KMPR.

The main program, DALG, is very short and merely needs a single card which controls access to the subprograms CREAD, FTEST, OPUT, and KMPR or indicates an exit from the program. The card read also specifies available options. The program CREAD reads, as input, cards describing the circuit to be tested. The circuit description is printed out and tested

for the presence of feedback loops. If no feedback loops are present, then control is returned to DALG. If a feedback loop is present, then control is not returned to DALG. If a feedback loop is present, then the logical gates of the circuit are partitioned into components each of which is a part of the same feedback loop. A gate with no feedback being considered as a component with only one member. The program is then terminated.

The subprogram FTEST assumes that CREAD has processed cards describing the circuit and attempts to find a test for gate or input ALPHA stuck at BETA (where BETA=0 or BETA=1). If such a test is found, then the pattern of inputs is checked via simulation for any fault on "the LIST"\* for which a test has not yet been found. As tests for specific faults are found, either by Roth's algorithm or by simulation, they are removed from "the LIST."

The subprogram OPUT will automatically attempt to find test patterns for all faults in the LIST. This is done by systematically submitting faults in the LIST to FTEST until the LIST is exhausted or until each fault has been submitted to FTEST.

The subprogram KMPR uses a sequence of test patterns to screen faulty circuits. The test patterns may come from any one of the following:

- 1) the output of FTEST
- 2) the output of OPUT
- 3) card input
- 4) a previous execution of KMPR.

The faulty circuits to be screened may be specified or may be randomly generated.

---

\*"the LIST" of faults initially consists of all input and gates stuck at 0 and stuck at 1. For example, suppose the circuit under consideration consists of two inputs I1 and I2 and one gate G1, then the LIST (of faults) would contain six (6) items namely: 1) I1 stuck at 0, 2) I1 stuck at 1, 3) I2 stuck at 0, 4) I2 stuck at 1, 5) G1 stuck at 0, and 6) G1 stuck at 1.

## INPUT

There are three (3) main types of input cards: 1) Control cards for DALG, 2) Circuit description cards for CREAD, and 3) KMPR specification cards. The control cards for DALG are shown in Table 1.

### CIRCUIT DESCRIPTION CARDS READ BY CREAD

The various gates and inputs in the circuit are identified by a two BCD character symbol (any two symbols may be used except a double blank which is used as a delimiter). The kinds of logical gates allowed are AND, NAND, OR, NOR, NOT, and EXCLUSIVE-OR and these functions are denoted as shown below.

<u>Type of Gate</u>	Function Symbol (Two characters)
AND	Ab*
NAND	NA
OR	OR
NOR	NO
NOT	Nb*
EXCLUSIVE OR	EO

\* (b= blank)

The first card read by CREAD is a card which contains the symbols of each of the inputs; Cols. 1-2 contain the first input symbol, Cols. 3-4 contain the second input symbol, Cols. 4-6 contain the third input symbol, Cols. 7-8 contain the fourth input, etc.

The next card read by CREAD contains a description of a gate in the circuit; the format of the card is as follows:

Cols. 1-2	Symbolic designation of gate
Cols. 3-4	Function symbol of gate
Cols. 5-6	Symbolic designation of first input
Cols. 6-7	Symbolic designation of second input
Etc.	

TABLE 1  
DESCRIPTION OF DALG CONTROL CARDS

Type	1st Entry	2nd Entry	3rd Entry	Effect or Purpose
0	Col. 5=0	Cols. 9-10 = two alpha - numeric characters	-	Will cause program to print "Programmed Exit" "ALPHA" = (Cols. 9-10 printed out) and stop
1	Col. 5=1	-	-	Will transfer control to CREAD
2	Col. 5=2	Cols. 9-10 = ALPHA	Col. 10 = 0 or 1	Transfers control to FTST and attempts to find a test for gate or input alpha stuck at zero or one
3	Col. 5=3	-	-	Transfers control to OPUT
4	Col. 5=4	Cols. 9-10 = ALPHA	Cols. 11-15 = BETA	Transfers control to KMPR and screens faulty circuits. For description of function of ALPHA and BETA see section on KMPR input.

There is one such card for each gate in the circuit. There is no required order for these cards described. Following all of the cards describing the gates in a circuit there must be a blank card and then a card listing the gates or inputs which are also used as outputs. This card has the format:

Cols. 1-2	First output symbol
Cols. 3-4	Second output symbol
Cols. 5-6	Third output symbol
Etc.	

### KMPR Specification Cards

The subprogram KMPR uses a sequence of test patterns to screen for fault detection. Therefore, the data needed by this program should specify the test patterns and the faulty circuits. There are two classes of KMPR specification cards and several types of cards within each class. They are described below.

### Test Pattern Cards

Type TP1 - Title card - will be reproduced as a part of the heading describing a test pattern. The first two columns of the card may not both be blank, otherwise any message is acceptable.

Type TP2 - Specifies the values of each input in the test pattern.

<u>Column</u>	<u>Contents</u>
1-2	Input Symbol
3	Value (0 or 1)
4-5	Input Symbol
6	Value (0 or 1)
Etc.	

There are no more than 26 input symbols and corresponding values on a card. If there are more than 26 inputs to the circuit, then after

specifying 26 input symbols and the corresponding values on a card, the following card will contain the remaining input symbols and values starting in column 1. There is no required order for the inputs on the card, but all of the inputs in the circuit must be specified or there will be an error stop.

Type TP3 - Indicates the end of sequence of test pattern cards. Columns 1-6 must be blank. Otherwise, the card may contain any legitimate characters in columns 7-80.

#### Faulty Circuit Cards

Type FC1 - Specifies logical faults in a circuit.

<u>Column</u>	<u>Contents</u>
1-2	Faulty Gate or Input Symbol
3	Stuck-at-value (0-1)
4-5	Faulty Gate or Input Symbol
6	Stuck-at-value
Etc.	

N.B. Faulty Inputs must be listed before faulty gates.

There are no more than 26 faults allowed in a single circuit.

Type FC2 - Indicates end of sequence of type FC1 cards. Columns 1-2 of this card must be blank; otherwise, any legitimate characters may be punched in columns 3-80.

Type FC3 - Specifies number of faulty circuits with BETA pseudo-random faults or less per circuit. Also specifies status of random number generator. The format of this card is:

<u>Column</u>	<u>Contents</u>
1-5 (right justified)	N = the number of faulty circuits
6-10(right justified)	N1 - If N1 is positive, the pseudo-random number generator is initialized and called N1 times. If N1 is not positive, then the status of the pseudo-random number generator is unchanged.

### The order of appearance of KMPR specification cards in an input deck

The transfer to KMPR from DALG is caused by a type 4 DALG control card. The order of KMPR specification cards depends upon the two parameters ALPHA and BETA specified on the type 4 DALG control card.

Case 1A - ALPHA = two blank characters.

There are no test pattern cards (test patterns have already been specified by other parts of the program such as FTEST, OPUT, or previous calls to KMPR).

Case 1B - ALPHA two blank characters

For each test pattern, there are in the order listed:

- 1) One type TP1 card
- 2) One (26 or less inputs) or two type TP2 card(s).

At the end of the sequence of types TP1 and TP2 cards, there must be a type TP3 card.

Case IIA - BETA = 0

There is for each faulty circuit a type FC1 card. After all of the type FC1 cards, there must be a type FC2 card.

Case IIB - BETA is positive. There is one only type FC3 card. In any case, test pattern cards (if any) always proceed faulty circuit cards.

### OUTPUT

The output of DALG may be classified as 1) CREAD output, 2) FTEST output, 3) OPUT output, 4) KMPR output, 5) exit statement, and 6) error message.

#### 1) CREAD Output

- a. "ENTER CREAD"

- b. The symbols used for inputs in the circuit and listed along with the corresponding numeric designation in the program.
- c. The number of logical gates in the circuit is printed.
- d. The gate description cards are listed in the order of appearance in the input deck.
- e. If there is no feedback in the circuit, then a line is printed for each logical gate in the circuit. Each line contains the items listed below in the order listed:
  - i) Internal (Program) number of the gate
  - ii) Number of inputs to the gate
  - iii) Gate Symbol
  - iv) A number corresponding to the logical function of the gate.  
NAND=1, NOR=2, AND=3, OR=4, NOT=5, EXCLUSIVE OR=6
  - v) The internal numbers which correspond to the gate inputs with circuit inputs being denoted by negative integers.
  - vi) The last item on each line is either 0 or -1. It is equal to -1 if the gate is a circuit output; otherwise, it is set equal to 0.
- f. If there is feedback in the circuit, then the gates of the circuit are partitioned into strongly connected components and the contents of each strongly connected component are listed. In this case, the program will not return control to DALG but rather will exit.

2) FTEST Output - Since FTEST attempts to find a test for gate or input ALPHA stuck at BETA, there are two possibilities:

- a. The attempt to find a test fails in which case the output is:

"TEST FOR ALPHA STUCK AT BETA DOES NOT EXIST"

b. A test is found in which case the output is:

i) The values of each input and logical gate output are listed for the failure free circuit under the heading:

"TEST CUBE FOR FAILURE FREE GATE"

ii) The values of each input and logical gate output are listed for the circuit with the failure specified under the heading:

"TEST CUBE FOR ALPHA STUCK AT BETA"

iii) If the test cube cited in i) above will also detect any other failures on the LIST which have not been deleted by this or other preceeding tests, then the test cube of the failing circuit is also printed out under an appropriate heading.

Note: Since the output of FTEST depends upon the failures in the LIST which have not been deleted, it (the output) is dependent to some extent upon previous input.

3) OPUT Output - Since OPUT scans the LIST for failures to be detected and requests that FTEST attempt to find a test for such failures, then the output of OPUT will be a sequence of FTEST outputs.

4) KMPR Output - is as follows:

a. "THE FOLLOWING ITEST BOOLEAN CUBES ARE TO BE USED AS TEST PATTERNS FOR SCREENING FAULTS" (where ITEST is the number of test patterns).

b. For each of the ITEST test patterns, there is printed:

i) "TEST PATTERN NO. "j" ---HEADING---"

ii) Boolean cube of test pattern

c. If BETA is zero, then for each faulty circuit, the following is printed:

i) "CIRCUIT WITH FAULTS SPECIFIED BELOW"

- ii) Specification of which gates and/or inputs are stuck and the values at which they are stuck.
- iii) For each test pattern, one of the following is printed:
  - $\alpha$ ) "TEST PATTERN "j" WILL NOT DETECT THE FAULTS AS GIVEN"
  - $\beta$ ) "TEST PATTERN "j" WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW."

The Boolean cube of the faulty circuit is then listed.

d. If BETA is positive, then

- i) If N1 (see KMPR input) is positive, the following message is printed:

"PSUEDO-RANDOM NUMBER GENERATOR INITIALIZED AND CALLED "N1" TIMES."

If N1 is not positive, then this message is deleted and the status of the pseudo random number generator is unchanged.

- ii) "THE DATA GIVEN BELOW CONSISTS OF "N" CIRCUITS EACH OF WHICH HAS "BETA" PSEUDO-RANDOMLY GENERATED FAULTS EACH IS TO BE SCREENED BY "ITEST" TEST PATTERNS" (where N and BETA are defined in the section on the input to KMPR).

- iii) For each faulty circuit, the following is printed:
 

"PSEUDO-RANDOM FAULTS ARE" followed by a listing of the pseudo-random faults.

The output described in c. iii) is repeated.

5) The exit statement is always "PROGRAMMED EXIT, ALPHA = A" where A = the two alphanumeric characters in cols. 9-10 of the DALG control card which caused the exit.

6) Error Message

<u>TYPE</u>	<u>DESCRIPTION</u>
1	Error on card describing circuit inputs (first card read by CREAD)
2	Too many cards describing gates
3	Improper functional description symbol on a gate description card (must be one of the following: 1) NA, 2) NO, 3) Ab, 4) OR, 5) Nb, 6) EO (b=blank))
4	Improper symbol for an input to a gate on a gate description card
5	More than 20 inputs on a gate description card
6	Not used
7	No inputs to a gate
8	Symbol not a gate output
9	Error in <u>Consistency Phase</u>
10	Error in FASTRAN records
11-12	Error in program logic
13	Machine error
14	Test cube is not a test for specified fault
15	No test pattern cards in KMPR input
16	Error in gate on input symbol specifying a fault
17	Option card in error
18	Not used
19	Not used
20	Error in program logic

### REFERENCES

- 1) Roth, J. P., Bouricius, W. G., and Schneider, P. R.

"Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEE Transactions on Electronic Computers, Vol. EC-16, No. 5, pp 567-579.

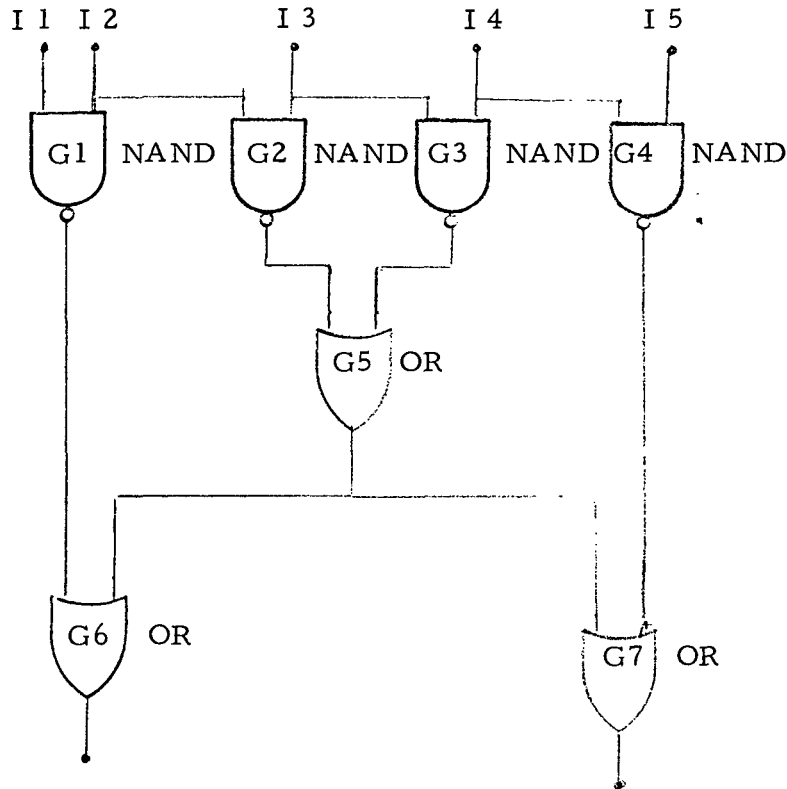
- 2) Chang, H. Y., Manning, E., and Metze, G.

"Fault Diagnosis of Digital Systems," 1970  
Willey - Interscience

## SAMPLE PROBLEM

We have drawn below two circuits for which test cubes will be developed by DALG

Circuit #1



Inputs: I1, I2, I3

Outputs: G6, G7

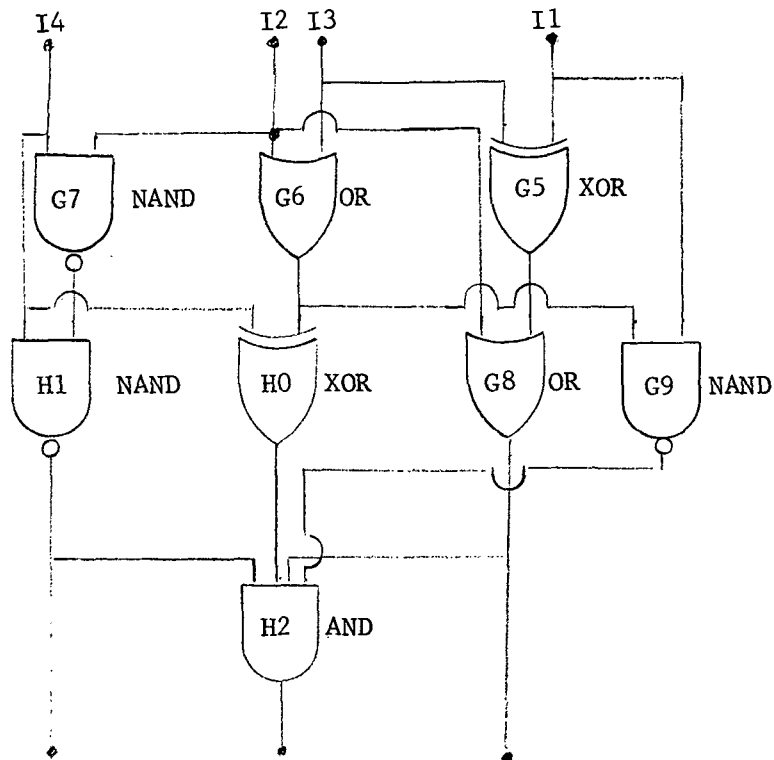
Find test for I1 stuck at 1 and test for G4 stuck at 1.

Check the resulting test pattern as screening tests for the following fault conditions.

1. I1 stuck at 1 and G5 stuck at 0
2. G6 and G7 both stuck at 1
3. 5 circuits each with 3 pseudo random faults

# S A M P L E   P R O B L E M

Circuit #2



Inputs:     I1, I2, I3, I4

Outputs:    H1, H2, G8

Find tests for each input or logical gate stuck at 0 or stuck at 1.

Consider the test patterns

1. All inputs are 0
2. All inputs are 1

Will these test patterns detect G6 and G5 stuck at 0?

The sample data sheets attached show the format of the input necessary to obtain the desired tests. The attached printed output shows that for the first circuit, the test for I1 stuck at 1 is also a test for:

- 1) I5 stuck at - 1
- 2) G1 stuck at - 0
- 3) G4 stuck at - 0
- 4) G6 stuck at - 0
- 5) G7 stuck at - 0

It should be noted that the additional tests which result from inputs to FTEST will not be produced again. That is, since a test for I5 stuck at 1 has already been produced, then this fault condition has been deleted from the LIST and no further attempts will be made to find tests for it unless specifically requested by a DALG control card.

If an additional line is added to the first circuit so that the output of gate G6 is now an input to gate G1 then this circuit now has feedback and if an attempt to find tests for faults is made, the result will be testing the strong components of the circuit. This output is also shown.

# Data Sheet for Sample Problems

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
				1																				
I 1	I 2	I 3	I 4	I 5																				
G 1	N A	I 1	I 2																					
G 2	N A	I 2	I 3																					
G 3	N A	I 3	I 4																					
G 4	N A	I 4	I 5																					
G 5	Ø	R	G 2	G 3																				
G 6	Ø	R	G 1	G 5																				
G 7	Ø	R	G 5	G 4																				
G 6	G 7																							
			2				I 1						0											
			2				G 4						1											
			4																					
G 6	G 1	5	0																					
			4										3											
			5						2															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# Data Sheet for Sample Problems

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
				1																				
I 1	I 2	I 3	I 4																					
G 5	E 0	I 1	I 3																					
G 6	O R	I 2	I 3																					
G 7	N A	I 4	I 2																					
G 8	O R	G 5	I 2																					
G 9	N A	I 1	G 6																					
H 1	N A	I 4	G 7																					
H 0	E 0	I 4	G 6																					
H 2	A	H 1	H 0	G 9	G 8																			
H 1	H 2	G 8																						
		3																						
		4						X						0										
A L L		I N P U T S		A R E	Z E R O																			
I 1	0	I 2	0	I 3	0	I 4	0																	
A L L		I N P U T S		A R E	O N E																			
I 4	1	I 3	1	I 2	1	I 1	1																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

## Data Sheet for Sample Problems

[illegible]



```

I1 1 I2 1 I3 0 I4 1 I5 0
G1 0 G2 1 G3 1 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR I4 STUCK AT 0
I1 1 I2 1 I3 1 I4 0 I5 0
G1 0 G2 0 G3 1 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR G1 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1
TEST CUBE FOR G2 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 1 G3 0 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR G3 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 0 G3 1 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR G5 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 0 G3 0 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR G6 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1
TEST CUBE FOR FAILURE FREE GATE

```

```

I1 0 I2 1 I3 1 I4 1 I5 1
G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 0
TEST CUBE FOR I5 STUCK AT 0
I1 0 I2 1 I3 1 I4 1 I5 0
G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1
TEST CUBE FOR G4 STUCK AT 1
I1 0 I2 1 I3 1 I4 1 I5 1
G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1
TEST CUBE FOR G7 STUCK AT 1
I1 0 I2 1 I3 1 I4 1 I5 1
G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 1
TEST CUBE FOR FAILURE FREE GATE

```

```

I1 1 I2 0 I3 1 I4 1 I5 0
G1 1 G2 1 G3 0 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR I2 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1
TEST CUBE FOR FAILURE FREE GATE

```

```

I1 1 I2 1 I3 0 I4 1 I5 0
G1 0 G2 1 G3 1 G4 1 G5 1 G6 1 G7 1
TEST CUBE FOR I3 STUCK AT 1
I1 1 I2 1 I3 1 I4 1 I5 0
G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1
TEST CUBE FOR G5 STUCK AT 0
I1 1 I2 1 I3 0 I4 1 I5 0
G1 0 G2 1 G3 1 G4 1 G5 0 G6 0 G7 1
TEST CUBE FOR FAILURE FREE GATE

```

```

I1 1 I2 1 I3 1 I4 0 I5 0

```

G1	0	G2	0	G3	1	G4	1	G5	1	G6	1	G7	1
TEST CUBE FOR I4 STUCK AT 1													
I1	1	I2	1	I3	1	I4	1	I5	0				
G1	0	G2	0	G3	0	G4	1	G5	0	G6	0	G7	1
TEST CUBE FOR G3 STUCK AT 0													
I1	1	I2	1	I3	1	I4	0	I5	0				
G1	0	G2	0	G3	0	G4	1	G5	0	G6	0	G7	1
TEST CUBE FOR FAILURE FREE GATE													
.													
I1	0	I2	1	I3	1	I4	1	I5	1				
G1	1	G2	0	G3	0	G4	0	G5	0	G6	1	G7	0
TEST CUBE FOR G4 STUCK AT 1													
I1	0	I2	1	I3	1	I4	1	I5	1				
G1	1	G2	0	G3	0	G4	1	G5	0	G6	1	G7	1

THE FOLLOWING 7 BOOLEAN CUBES ARE TO BE USED AS TEST PATTERNS FOR SCREENING FAULTS

TEST PATTERN NO. 1 TEST FOR I1 STUCK AT 1  
I1 0 I2 1 I3 1 I4 1 I5 0

G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1

TEST PATTERN NO. 2 TEST FOR I1 STUCK AT 0  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1

TEST PATTERN NO. 3 TEST FOR I5 STUCK AT 0  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 0

TEST PATTERN NO. 4 TEST FOR I2 STUCK AT 1  
I1 1 I2 0 I3 1 I4 1 I5 0

G1 1 G2 1 G3 0 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 5 TEST FOR I3 STUCK AT 1  
I1 1 I2 1 I3 0 I4 1 I5 0

G1 0 G2 1 G3 1 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 6 TEST FOR I4 STUCK AT 1  
I1 1 I2 1 I3 1 I4 0 I5 0

G1 0 G2 0 G3 1 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 7 TEST FOR G4 STUCK AT 1  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 0

CIRCUIT WITH FAULTS SPECIFIED BELOW

I1 S.A. 1, G5 S.A. 0,

TEST PATTERN 1 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW

I1 1 I2 1 I3 1 I4 1 I5 0

G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 2 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW

I1 1 I2 1 I3 1 I4 1 I5 1  
G1 0 G2 0 G3 0 G4 0 G5 0 G6 0 G7 0

TEST PATTERN 4 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 5 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 0 I4 1 I5 0  
G1 0 G2 1 G3 1 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 6 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 0 I5 0  
G1 0 G2 0 G3 1 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 1  
G1 0 G2 0 G3 0 G4 0 G5 0 G6 0 G7 0

CIRCUIT WITH FAULTS SPECIFIED BELOW

G6 S.A. 1, G7 S.A. 1.

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 0  
G1 0 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 1  
G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 1

TEST PATTERN 4 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 5 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 6 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 1  
G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 1

THE FOLLOWING 7 BOOLEAN CUBES ARE TO BE USED AS TEST PATTERNS FOR SCREENING FAULTS

TEST PATTERN NO. 1 TEST FOR I1 STUCK AT 1  
I1 0 I2 1 I3 1 I4 1 I5 0

G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1

TEST PATTERN NO. 2 TEST FOR I1 STUCK AT 0  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1

TEST PATTERN NO. 3 TEST FOR I5 STUCK AT 0  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 0

TEST PATTERN NO. 4 TEST FOR I2 STUCK AT 1  
I1 1 I2 0 I3 1 I4 1 I5 0

G1 1 G2 1 G3 0 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 5 TEST FOR I3 STUCK AT 1  
I1 1 I2 1 I3 0 I4 1 I5 0

G1 0 G2 1 G3 1 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 6 TEST FOR I4 STUCK AT 1  
I1 1 I2 1 I3 1 I4 0 I5 0

G1 0 G2 0 G3 1 G4 1 G5 1 G6 1 G7 1

TEST PATTERN NO. 7 TEST FOR G4 STUCK AT 1  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 0

THE DATA GIVEN BELOW CONSISTS OF 5 CIRCUITS EACH OF WHICH HAS 3 PSEUDO-RANDOMLY GENERATED FAULTS  
EACH IS TO BE SCREENED BY 7 TEST PATTERNS

PSEUDO RANDOM FAULTS ARE  
I2 S.A. 1, I5 S.A. C, G7 S.A. 1,

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 0

G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1

TEST PATTERN 4 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 5 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 6 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 0

G1 1 G2 0 G3 0 G4 1 G5 0 G6 1 G7 1

PSEUDO RANDOM FAULTS ARE  
I4 S.A. 1, G7 S.A. 1, G5 S.A. C,

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 1

TEST PATTERN 4 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 5 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 0 I4 1 I5 0

G1 0 G2 1 G3 1 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 6 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 0 G2 0 G3 0 G4 1 G5 0 G6 0 G7 1

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 1 I4 1 I5 1

G1 1 G2 0 G3 0 G4 0 G5 0 G6 1 G7 1

PSEUDO RANDOM FAULTS ARE  
G6 S.A. 1, I2 S.A. 0, I1 S.A. 0,

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 0 I3 1 I4 1 I5 0

G1 1 G2 1 G3 0 G4 1 G5 1 G6 1 G7 1

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 0 I3 1 I4 1 I5 1

G1 1 G2 1 G3 0 G4 0 G5 1 G6 1 G7 1

TEST PATTERN 4 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 5 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 6 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 0 I3 1 I4 1 I5 1

G1 1 G2 1 G3 0 G4 0 G5 1 G6 1 G7 1

PSEUDO RANDOM FAULTS ARE  
G6 S.A. 1, I3 S.A. 0, G2 S.A. 1,

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 0 I4 1 I5 0

G1 C G2 1 G3 1 G4 1 G5 1 G6 1 G7 1

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 0 I4 1 I5 1

G1 1 G2 1 G3 1 G4 C G5 1 G6 1 G7 1

TEST PATTERN 4 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 5 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 6 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 7 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 0 I2 1 I3 0 I4 1 I5 1

G1 1 G2 1 G3 1 G4 C G5 1 G6 1 G7 1

PSEUDO RANDOM FAULTS ARE  
I1 S.A. 1, G7 S.A. 0, G3 S.A. C,

TEST PATTERN 1 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 C G2 0 G3 C G4 1 G5 C G6 0 G7 C

TEST PATTERN 2 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 0

G1 C G2 0 G3 0 G4 1 G5 C G6 0 G7 C

TEST PATTERN 3 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 1 I4 1 I5 1

G1 0 G2 C G3 C G4 C G5 0 G6 0 G7 C

TEST PATTERN 4 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 0 I3 1 I4 1 I5 0

G1 1 G2 1 G3 C G4 1 G5 1 G6 1 G7 C

TEST PATTERN 5 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW  
I1 1 I2 1 I3 C I4 1 I5 0

G1 C G2 1 G3 C G4 1 G5 1 G6 1 G7 C

G1 0 G2 0 G3 3 G4 1 G5 0 G6 3 G7 3

G1 0 G2 0 G3 0 G4 0 G5 0 G6 0 G7 0

```

1 2 3 4
I1 I2 I3 I4
NGTS = 8
G5 E0 I1 I3
G6 G0 I2 I3
G7 NA I4 I2
G8 G0 G5 I2
G9 NA I1 G6
H1 NA I4 G7
H0 E0 I4 G6
H2 A H1 H0 G9 G8
1 2 G5 6 -1 -3 0
2 2 G6 4 -2 -3 0
3 2 G7 1 -4 -2 0
4 2 G8 4 1 -2 -1
5 2 G9 1 -1 2 0
6 2 H1 1 -4 3 -1
7 2 H0 6 -4 2 0
8 4 H2 3 6 7 5 4 -1
TEST CUBE FOR FAILURE FREE GATE

```

[illegible]

I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	0	G9	1	H1	1	H0	1	H2	0	
TEST CUBE FOR G9 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	0	H1	1	H0	1	H2	0	
TEST CUBE FOR H1 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	0	H0	1	H2	0	
TEST CUBE FOR H0 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	0	H2	0	
TEST CUBE FOR H2 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	1	H2	0	
TEST CUBE FOR FAILURE FREE GATE																
I1	1	I2	0	I3	1	I4	0									
G5	0	G6	1	G7	1	G8	0	G9	0	H1	1	H0	1	H2	0	
TEST CUBE FOR I1 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	1	H2	1	
TEST CUBE FOR I2 STUCK AT 1																
I1	1	I2	1	I3	1	I4	0									
G5	0	G6	1	G7	1	G8	1	G9	0	H1	1	H0	1	H2	0	
TEST CUBE FOR G5 STUCK AT 1																
I1	1	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	0	H1	1	H0	1	H2	0	
TEST CUBE FOR G8 STUCK AT 1																
I1	1	I2	0	I3	1	I4	0									
G5	0	G6	1	G7	1	G8	1	G9	0	H1	1	H0	1	H2	0	
TEST CUBE FOR H2 STUCK AT 1																
I1	1	I2	0	I3	1	I4	0									
G5	0	G6	1	G7	1	G8	0	G9	0	H1	1	H0	1	H2	1	
TEST CUBE FOR FAILURE FREE GATE																
I1	0	I2	0	I3	0	I4	0									
G5	0	G6	0	G7	1	G8	0	G9	1	H1	1	H0	0	H2	0	
TEST CUBE FOR I3 STUCK AT 1																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	1	H2	1	
TEST CUBE FOR FAILURE FREE GATE																
I1	0	I2	0	I3	1	I4	1									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	0	H0	0	H2	0	
TEST CUBE FOR I4 STUCK AT 0																
I1	0	I2	0	I3	1	I4	0									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	1	H2	1	
TEST CUBE FOR G7 STUCK AT 0																
I1	0	I2	0	I3	1	I4	1									
G5	1	G6	1	G7	0	G8	1	G9	1	H1	1	H0	0	H2	0	
TEST CUBE FOR H1 STUCK AT 1																
I1	0	I2	0	I3	1	I4	1									
G5	1	G6	1	G7	1	G8	1	G9	1	H1	1	H0	0	H2	0	
TEST CUBE FOR FAILURE FREE GATE																

```

I1 1 I2 1 I3 1 I4 0
G5 0 G6 1 G7 1 G8 1 G9 0 H1 1 H0 1 H2 0
TEST CUBE FOR I2 STUCK AT 0
I1 1 I2 0 I3 1 I4 0
G5 0 G6 1 G7 1 G8 0 G9 0 H1 1 H0 1 H2 0
TEST CUBE FOR G9 STUCK AT 1
I1 1 I2 1 I3 1 I4 0
G5 0 G6 1 G7 1 G8 1 G9 1 H1 1 H0 1 H2 1
TEST FOR G6 STUCK AT 1 DOES NOT EXIST
TEST CUBE FOR FAILURE FREE GATE

I1 0 I2 1 I3 0 I4 1
G5 0 G6 1 G7 0 G8 1 G9 1 H1 1 H0 0 H2 0
TEST CUBE FOR G7 STUCK AT 1
I1 0 I2 1 I3 0 I4 1
G5 0 G6 1 G7 1 G8 1 G9 1 H1 0 H0 0 H2 0
TEST CUBE FOR H0 STUCK AT 1
I1 0 I2 1 I3 0 I4 1
G5 0 G6 1 G7 0 G8 1 G9 1 H1 1 H0 1 H2 1

```

THE FOLLOWING 2 BOOLEAN CUBES ARE TO BE USED AS TEST PATTERNS FOR SCREENING FAULTS

TEST PATTERN NO. 1 ALL INPUTS ARE ZERO

I1 0 I2 0 I3 0 I4 0

G5 0 G6 0 G7 1 G8 0 G9 1 H1 1 H0 0 H2 0

TEST PATTERN NO. 2 ALL INPUTS ARE ONE

I1 1 I2 1 I3 1 I4 1

G5 0 G6 1 G7 0 G8 1 G9 0 H1 1 H0 0 H2 0

CIRCUIT WITH FAULTS SPECIFIED BELOW

G6 S.A. 0, G5 S.A. 0,

TEST PATTERN 1 WILL NOT DETECT THE FAULTS AS GIVEN

TEST PATTERN 2 WILL DETECT THE FAULTS AS GIVEN. THE BOOLEAN CUBE OF THE FAULTY CIRCUIT SHOWN BELOW

I1 1 I2 1 I3 1 I4 1

G5 0 G6 0 G7 0 G8 1 G9 1 H1 1 H0 1 H2 1

## Appendix

The attached logic diagram shows an integrated circuit multigate-array logic element which serves as either a four bit adder/subtractor (Mode 1) or an eleven pole double throw logic switch (Mode 2). This device is one of the CMMAs being manufactured to JPL's specifications. The table shown below gives the number of simple faults for each mode.

Mode	No. of Inputs	No. of Logical Gates	No. of Simple Faults
1	22	64	172
2	24	44	136

A sequence of test patterns for this device were developed by DALG to detect simple faults and this sequence is denoted by Seq. dalg. Another sequence of test patterns were developed by a computer program written by others specifically for this logical device and this sequence is denoted by Seq. other.

This appendix briefly compares the two sequences of test patterns. The sequence of test patterns Seq. other were checked by DALG for

- 1) correctness of failure free output
- 2) ability to detect all possible simple faults and
- 3) ability to detect a sequence of 10 failures each with two failing gates or inputs.

The sequence of test patterns Seq. dalg which were produced to detect all simple faults were also checked against the same set of multiple faults. The results are summarized in the tables below.

Mode	Detect All Simple Faults		Detect Given Multiple Faults	
	Seq. dalg	Seq. other	Seq. dalg	Seq. other
1	Yes	No	Yes	Yes
2	Yes	Yes	Yes	Yes

The number of test patterns in Seq. dalg for Mode 1 and Mode 2 were 38 and 26 respectively. The sequence of test patterns Seq. other failed to detect one of the simple faults in Mode 1 and in addition one of the test patterns in Seq. other was found to be in error.

It is worth noting that the time required to prepare the input data to produce Seq. dalg was about 4 man hours and the cost of machine time was about \$25.



## ACKNOWLEDGMENT

The information documented in this report was compiled by Eugene Friedman of the Quality Assurance and Reliability Division of the Jet Propulsion Laboratory.